# 面向有效载荷高速数据流的数据处理方法

王静 [1,2]，王春梅 [1]，智佳 [1]，杨甲森 [1]，陈托 [1]

（1. 中国科学院国家空间科学中心，北京 100190；2. 中国科学院大学，北京 100049）

**摘要：**针对卫星有效载荷数传数据传输速度快、实时处理难等特点，提出一种面向有效载荷高速数据流的实时数据处理方法。该方法借鉴 MapReduce 的多线程并行模式，采用 hash 算法与归并排序算法相结合的方式，提高数据处理吞吐率，实现实时处理；采用基于 XTCE(XML Telemetry & Command Exchange)数据模型的参数解析算法，实现通用性。实验表明该方法能够满足有效载荷对数据处理的实时性和正确性的要求。

**关键词：**有效载荷高速数据流；数据处理；MapReduce；吞吐率；XTCE

**中图分类号：**TP311　　　　**文献标识码：**A

# Technology of Data Processing for High Speed Data Stream of Payload

WANG Jing[1,2], WANG Chun-Mei[1], ZHI Jia[1],YANG Jia-Sen[1], CHEN Tuo[1]

(1.National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China;

2.University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Aiming at the characteristics of high speed transmitting and difficult in real-time data processing for payload transmitted data, a method of data processing for high speed data stream of payload is presented. By drawing on multi-threaded parallel mode of MapReduce framework, hash algorithm and merge sort algorithm are combined to improve throughput rate of data processing and achieve real-time processing, a parameter parsing algorithm based on XTCE (XML Telemetry & Command Exchange) data model is employed to improve generality. Results of the experiments indicated the proposed method could meet data processing requirements of payload in real-time performance and validity.

**Key words:** high speed data stream of payload; data processing; MapReduce; throughput rate ; XTCE

---

作者简介：王静（1990-），女，山东威海人，博士研究生，研究方向为空间数据处理、自动测试技术；王春梅（1965-），女，北京人，研究员，博士生导师，研究方向为自动测试技术；智佳（1984-），男，山西太原人，工程师，硕士研究生，研究方向为空间数据处理；杨甲森（1979-），男，山东聊城人，副研究员，博士研究生，研究方向为自动测试技术；陈托（1986-），男，湖北天门人，工程师，硕士研究生，研究方向为数据库技术。E-mail：wangj711@yeah.net

## 0 引言

随着我国航天事业的快速发展，空间科学任务中有效载荷的数传数据量越来越大，数据类型更加复杂，而通信设备性能的日渐优化，使得数传数据的传输码率大幅提升。针对有效载荷高速数据流，即使所使用的计算机处理性能越来越高，采用常规的逐帧参数解析方法依然不能满足科学任务对实时性和正确性的要求，因此，研究面向有效载荷高速数据流的数据处理方法，逐渐成为航天数据处理的关键性课题。

目前国内外学者针对大规模数据的批量计算相关技术的研究相对成熟[1-3]，用于批处理计算的 MapReduce 框架得到广泛应用，然而现有的 Hadoop、Phoenix 等大规模数据处理技术，更偏向于处理永久性数据，不但不能满足大数据流式计算在实时性、吞吐率方面的要求[4]，更存在与已有数据处理模块无法兼容的问题。大数据流式计算的研究我国目前还属于起步状态，国外虽然已开发出一些流式计算系统，如 Twitter 的 Storm 系统，Yahoo 的 S4 系统等，但更适用于数据挖掘，对数据处理的正确性和实时性要求较低[5]。

针对有效载荷数据的处理，重点在于同步码识别、数据分包、拼接、参数数据提取、参数物理量转换、数据定标、修正、反演等一系列正确性要求较高的操作，这与目前已有的流式计算系统的应用有较大差别。此外，多型号多任务是未来航天事业的发展方向，数据处理方法需要能够具有良好的通用性，以适应多异构任务，并有效降低开发成本。鉴于此，设计一种适用于有效载荷高速数据流的通用数据处理方法十分必要。

## 1 有效载荷实时数据处理系统

有效载荷数传数据简称传输帧，其格式遵循国际空间数据系统咨询委员会（Consultative Committee for Space Data Systems, CCSDS）的高级在轨系统（Advanced Orbit System, AOS）标准[6]，示例结构如表 1。数据处理过程繁杂，主要分为两部分：预处理和参数解析，此外，通常须辅以参数入库系统保存数据，以及数据监视系统监视各部分的状态。系统结构如图 1 所示。

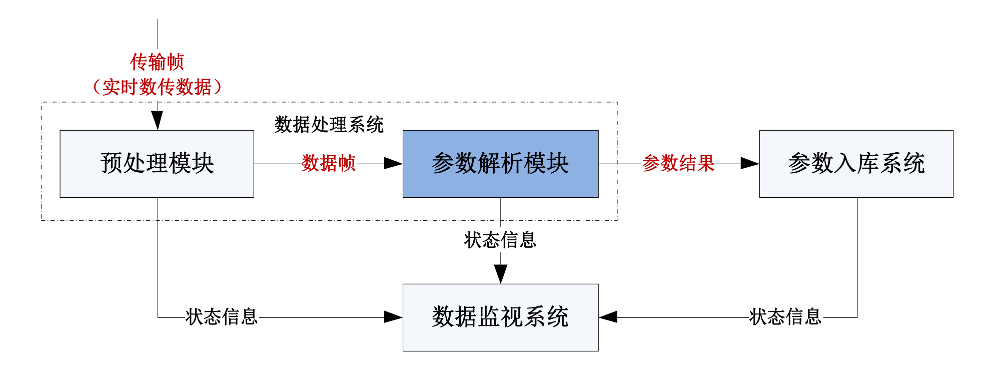



图 1 实时数据处理系统结构图

预处理模块对传输帧的数据域进行分包和组装处理，形成表 2 中的数据帧结构，处理较快，一般不存在速率瓶颈问题。参数解析模块对数据帧中的每一参数进行处理，数据帧的应用数据段长度不固定，工程应用中，应用数据的形式有几十种，每一种包含几十至几百个参数，参数越多，参数值转换越复杂，解析过程越耗时。

随着空间科学任务的发展，有效载荷数据量显著增多，对数据处理吞吐率的要求大幅提升。由于参数解析模块的处理过程复杂且耗时，常规的逐帧处理方法在处理有效载荷高速数据流时会产生严重的数据堆积，降低处理的实时性和正确性，甚至导致系统中断。针对这一问题，设计一种面向有效载荷高速数据流的数据处理方

法，该方法借鉴 MapReduce 模型分而治之的思想[7-9]，设计 hash 算法与归并排序算法相结合的多线程并行模式，实现对有效载荷高速数据流的实时数据处理，满足空间科学任务型号对吞吐率、实时性及正确性的要求。

表 1 传输帧结构

| 同步码 | 传输帧主导头 | | | | 传输帧数据域 | | 编码校验符号填充 |
|---|---|---|---|---|---|---|---|
| | 主信道标识符 | 虚拟信道标识符 | 虚拟信道传输帧计数 | 信令域 | B_PDU 位流数据 | | |
| 4B | 2B | | 3B | 1B | 2B | 884B | 128B |

表 2 数据帧结构

| 传输标签 | | | | | | | | | 数据段 |
|---|---|---|---|---|---|---|---|---|---|
| 同步头 | 卫星标识 | 消息类型 | 消息长度 | 系统时间 | 信源 | 信宿 | 计数 | 保留 | 应用数据 |
| 4B | 4B | 4B | 2B | 6B | 4B | 4B | 4B | 4B | 变长 |

## 2 关键技术

设计面向有效载荷高速数据流的处理方法，主要从两方面提升参数解析模块的性能：一是采用高效的处理模式，在不影响数据正确性的前提下，将逐帧处理改为并行处理；二是采用通用的解析算法，使其能够面向多任务，具有良好的通用性。

### 2.1 并行处理模式

MapReduce 批处理方法对于可划分的大规模数据处理任务，能够提供充分的并行计算语义，针对有效载荷高速数据流，借鉴其分而治之的思想，采用多线程并行的模式进行处理。数据处理架构如图 2 所示，可表述为三步：

（1） Hash 分流

通过识别同步码，将数据流切分为数据帧，以独立数据帧为对象，采用 Hash 方法进行等价映射，并分配到相应的 hash 桶中。常见的 hash 函数构造方法[10]有：

- 直接取余法：$Hash(x) = x \bmod p$，$p \le m$，$m$ 为 hash 桶的数目（或 hash 链表的长度）。
- 乘余取整法：$Hash(x) = \lceil m(\theta \times x\%1) \rceil (0 \le \theta \le 1)$，其中 $\theta \times x\%1$ 表示 $\theta \times x$ 的小数部分。
- 平方取中法：取平方后的中间几位数字。

为实现真正意义上的多线程并行，用于处理有效载荷高速数据流的 hash 桶的数目要小于 2 倍的计算机处理器核数。处理空间数据的计算机配置较高，一般配备有 32 及以上数目的核数，因此，hash 桶的数目一般可选取为十几个至几十个，采用直接取余法即简单易操作，又能使得数据均匀分布，是一种较优的选择。

（2） 数据帧处理

即实现数据帧的参数解析。为每个存放数据帧的 hash 桶分配一个固定的处理线程和结果缓存区，处理线程按 FIFO（First Input First Output）的方式从桶内获取数据帧，按数据帧格式规范将其解析为相应参数，最后将参数的内容存储到对应的结果缓冲区中。

（3） 归并

将各个结果缓存区中的参数结果按时间先后顺序进行归并，并将结果发送给实现参数入库的软件。鉴于该方法设计的特殊性，其排序过程可以不通过比较时间字段实现，而是从结果缓存区 1~n 中顺次取一个参数帧，即为按时间先后顺序。每个结果缓存区同样采取 FIFO 的队列式方式存取。

图 2 中数据接收缓存、数据帧缓存区以及处理结果缓存区均设计为环形队列的结构，可以使固定大小的内存空间反复使用，不需要进行动态的内存释放和分配。

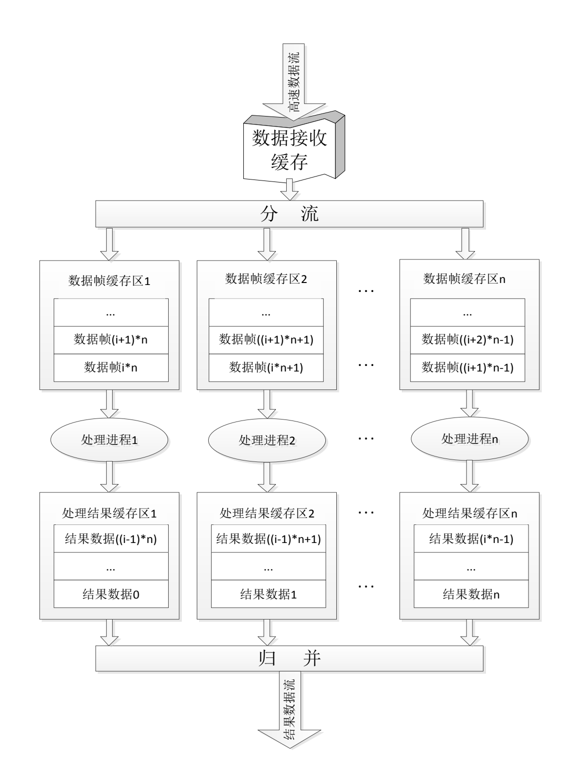



图 2  有效载荷高速数据流处理架构

## 2.2  数据帧处理

处理数据帧需要数据帧及其参数的结构信息，同时，解析的结果还需要满足用户对数据显示的需求。以往，数据帧处理过程需要针对不同任务的数据特点建立相应的数据库、开发特定的参数解析软件，工作量大，周期长，特别是不能满足未来多卫星多任务的航天需求。引入一种标准化的数据信息描述模型，可以使得参数解析模块能够不局限于单一航天任务，实现通用化。

CCSDS 推出的 XTCE（XML Telemetry & Command Exchange）是一种描述数据模型的国际化标准[11-13]，以 XML schema 文件的形式给出，顶层根节点为 SpaceSystem，包含 6 部分：Header（头文件）、Description（文件描述）、SpaceSystem（子系统）、CommandMetaData（遥控元数据）、TelemetryMetaData（遥测元数据）和 ServiceSet（服务集合）。参数解析模块所需的结构信息主要来源于 TelemetryMetaData 元素，其子元素包括 ParameterTypeSet（参数类型集合）、ParameterSet（参数集合）、ContainerSet（容器集合）等。

XTCE 覆盖范围广泛，为避免冗余，可针对需求进行裁剪。对于参数解析模块，通常需要参数名、起止位置、类型、转换公式等信息。系统初始化时，读取 XTCE 文件，解析并保存数据模型，以备各数据帧处理进程使用。数据帧处理进程的算法流程如图 2 所示。初始条件中，各变量所表示的含义分别为：

IsSyn：是否同步；MoveBit：所需移位数；LenofUnPrecess：未处理的数据长度；LenofLeft ：上一次处理后剩余的数据长度；UnitDataLength ：单位处理长度（数据帧长）。算法核心思想在于：设置两个缓冲区，缓冲区的大小设置为单次处理的最大长度的两倍，B1 用于处理，B2 用于备份，当 B1 寻找同步码产生移位时可用 B2 恢复 B1 中数据。新读取数据 LenofNewData 不长于单次处理的最大长度。每次读取固定长度数据，判断是否同步，同步则解析；不同步，则重新寻找同步码，从新同步码位置解析，若仍未找到同步码，则只保留该次处理数据末尾的同步码长度的数据，以确保不影响对下一次读取的新数据的同步状态识别。寻找同步码时采用按字节比较的方式，若一次比较未找到，则将 B1 缓冲区右移一位，再次比较，至多移 7 次结束。

该算法通过匹配 XTCE 文件中的数据模型进行参数解析，不局限单一结构，具有良好的通用性。

开始

初始条件：IsSyn = false; MoveBit = 0;
LenofUnPrecess = 0;LenofLeft = 0;
UnitDataLength=0;

申请两个缓冲区B1，B2。

读取定长数据：记录数据长度为LenofNewData；
将新数据追加到B2缓冲区中；
记录LenofUnPrecess = LenofNewData + LenofLeft，
TotalLengthToProcess（总长）=
LengthOfUnProcess。

LenofUnPrecess >=UnitDataLen + SysCodeLen（同步码长度）

是

将B2缓冲区中的数据恢复至B1中；

IsSys = True

否

将B2缓冲区中的数据恢复至B1中；
从B1中寻找同步头，记录移位数Movebit（若未找到同步头，Movebit=-1）

Movebit >=0

是

记录同步状态IsSyn =True

否

丢弃数据；
记录LengthOfUnProcess = SysCodeLen

将B2中未处理数据覆盖B2中原有数据；
记录LenofLeft = LengthOfUnProcess

接收缓存为空

是

结束

否

否

是

将B1缓冲区右移Movebit位；
获取同步码结束位置SynCodePos；
获取数据帧的长度，赋给UnitDataLength；
LenofUnPrecess = TotalLengthToProcess - SynCodePos；

LenofUnPrecess >=UnitDataLen + SysCodeLen（同步码长度）

是

该帧后的数据为同步头

是

LenofUnPrecess -= UnitDataLen ；
合法帧计数+1；
匹配数据模型解析：
获取数据帧标识（读取指定位置的bit位），
查找该标识对应的参数结构；
依次截取参数并进行物理量转换；
将结果存入对应结果缓冲区；

否

否

LengthOfUnProcess = TotalLengthToProcess - SynCodePos -1; 非法帧计数+1；
记录失步状态：IsSyn = false;

IsSys = True

否

是

Movebit>0&&
LenofUnProcess<TotalLenToProcess

是

LenofUnProcess+=1;

图 2 数据帧处理进程流程图

## 3 系统实现与验证

采用 Visual Studio 2012 工具开发了面向有效载荷高速数据流的数据处理系统，将该系统部署在 linux 系统的服务器上。应用某型号卫星的数传数据对系统进行测试，该卫星要求实时处理吞吐率不小于 150 $Mbps$。搭建系统测试验证平台如图 3 所示，在模拟发送计算机上安装模拟发送程序，通过网络按照指定码率发送数传数据到数据处理服务器，数据处理服务器上部署数据处理系统，系统中各模块间（包含数据库）使用万兆网互联，系统状态信息通过千兆网发送给数据处理监视系统。

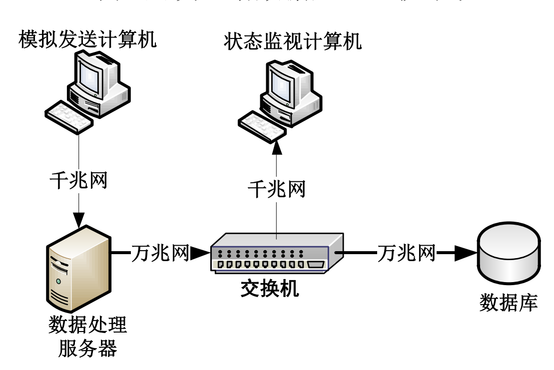



图 3 系统测试验证平台

测试表明，面向有效载荷高速数据流的处理方法性能优于常规方法，其测试比较如表 3 所示，需要说明的是，常规方法是事后处理方式，当采用实时处理方式时，码率高于 12.1Mbps 会导致接收缓存溢出进而导致处理中断。

此外，通用的参数解析算法，使得参数解析模块能够面向多任务，当数据结构发生变化时，只需更新 XTCE 文件，无需更改程序，降低开发成本。

表 3 两种方法测试结果比较

| 比较 | 常规方法 | 高速数据流处理方法 |
|---|---|---|
| 处理模式 | 单线程逐帧处理模式 | 多线程并行处理模式 |
| 服务器配置 | 4×16 核，64G 内存 | 4×16 核，64G 内存 |
| 模拟发送速率 | / | 145Mbps、155Mbps |
| 最大吞吐率 | 12.1Mbps | >155Mbps（16 个线程） |
| 处理方式 | 事后 | 实时 |
| 合法帧计数 | 26624 | 26624 |

## 4 结束语

本文设计并实现了面向有效载荷高速数据流的数据处理系统，解决了常规方法无法满足较高数据处理吞吐率要求的问题，并能够面向多型号多任务，具有良好的通用性。该系统经过测试验证，可以实现目前对有效载荷高速数据流的处理需求，为工程应用提供参考。

## 参考文献


[1] WANG Yuanzhuo, JIN Xiaolong,CHENG Xueqi. Network big data:Present and future[J]. Chinese Journal of Computers. 2013,36(6):1125-1138(in Chinese).[王元卓,靳小龙,程学旗.网络大数据:现状与展望[J].计算机学报.2013,36(6): 1125-1138.]

[2] YANG D, RUNDENSTEINER EA, WARD M. Mining neighbor-based patterns in data streams. Information Systems,2013,38(3):331-350.

[3] LIM L, MISRA A, MO TL. Adaptive data acquisition strategies for energy-efficient, smartphone-based, continuous processing of sensor streams. Distributed and Parallel Databases, 2013,31(2):321-351.

[4] QI Kaiyuan, ZHAO Zhuofeng, FANG Jun, et al. Real-time processing for high speed data stream over large sacle data. Chinese Journal of Computers[J]. 2013, 35(3):477-490(in Chinese). [亓开元,赵卓峰,房俊等.针对高速数据流的大规模数据实时处理方法.计算机学报[J].2012.3,35(3): 477- 490.]

[5] SUN Dawei, ZHANG Guangyan, ZHENG Weimin. Big data stream computing: technologies and instances. Journal of software, 2014,25(4):839-862(in Chinese).[孙大为,张广艳,郑纬民.大数据流式计算：关键技术及系统实例.软件学报[J].2014，25(4):839-862.]

[6] Recommended Standard CCSDS 732.0-B-3. AOS SPACE DATA LINK PROTOCOL[S]. 2015.09.

[7] ANAND R, JEFFREY D U. Mining of massive datasets[M].Beijing:The people’s posts and telecommunications press.2012.

[8] DEAN J, GHEMAWAT S. MapReduce: Simplified data processing on large clusters. ACM Comm- unication, 2008, 51(1):107-113.

[9] 李玉林,董晶.基于 Hadoop 的 MapReduce 模型的研究与改进[J].计算机工程与设计,2012,33(8):3110-3116.

[10] YAN Weimin, WU Weimin. Data structure (C language)[M]. Beijing: Tsinghua university press,[严蔚敏,吴伟民.数据结构（C 语言版）

[M].北京:清华大学出版社,2006:253-362.]

[11] Recommended Standard CCSDS 660.0-B-1. XML TELEMETRIC AND COMMAND EXCHANGE(XTCE) [S]. 2007.10.

[12] QU Yi, LIU Yurong, ZUO Jiangtao,et al. Study on the architecture of telemetry processing software based on XTCE[J]. Journal of Spacecraft TT&C Technologty. 2012,31(1):60-64(in Chinese). [曲艺,刘玉荣,左江涛等. 基于XTCE标准的遥测数据处理软件架构研究[J]. 飞行器测控学报.2012,31(1):60-64.]

[13] Cao Yujuan, Wang Jun, Ou Yujun, et al. Review of CCSDS XTCE[J]. Journal of Spacecraft TT&C Technology,2012,31(suppl.):38-42(in Chinese).[曹玉娟,王军,欧余军,等. CCSDS XTCE 研究综述[J]. 飞行器测控学报,2012,31(增)：38-42.]